

UriVersioningOptions

August 18, 2026

UriVersioningOptions

Kind: Interface

Source: `packages/common/interfaces/version-options.interface.ts`

Part of: [Common](#)

`UriVersioningOptions` configures URI-based API versioning. It requires the versioning strategy to be `VersioningType.URI` and optionally customizes—or disables—the URI prefix used before the version segment.

Properties

Property	Type
<code>type</code>	<code>VersioningType.URI</code>
<code>prefix</code>	<code>`string</code>

Diagram

```
graph LR
  A[Application Versioning Configuration] --> B[UriVersioningOptions]
  B --> C["type: VersioningType.URI"]
  B --> D["prefix: string | false"]
  C --> E[Version read from request URI]
  D --> F[Generated route path]
  E --> F
```

Usage

```
import { VersioningType } from '@nestjs/common';
import type { UriVersioningOptions } from '@nestjs/common';

const versioningOptions: UriVersioningOptions = {
```

```

type: VersioningType.URI,
prefix: 'v',
};

// Produces routes such as: /v1/users
app.enableVersioning(versioningOptions);

```

AI Coding Instructions

- Always set `type` to `VersioningType.URI` ; this interface is only valid for URI-based versioning.
- Use a string `prefix` such as `'v'` when versioned routes should look like `/v1/resource` .
- Set `prefix: false` when the version should appear without a prefix, such as `/1/resource` .
- Pass these options to the application's versioning configuration, typically through `app.enableVersioning()` .
- Keep the configured prefix consistent with API gateway routes, documentation, and client URL construction.

How it works

`UriVersioningOptions` is a public TypeScript interface for selecting URI-based API versioning. Its required discriminant is `type: VersioningType.URI` ; its only URI-specific setting is the optional `prefix` , typed as `string | false` .

[packages/common/interfaces/version-options.interface.ts:45-58](#) `VersioningType.URI` is the `URI` member of the versioning-type enum. [packages/common/enums/version-type.enum.ts:4-8](#)

- Set `prefix` to a string to prepend that string to a route's version segment. When absent, route construction selects `'v'` ; when `false` , it selects an empty prefix. [packages/core/router/route-path-factory.ts:86-96](#)
- URI versioning alters generated paths only when route metadata has a truthy method or controller version and the configured type is `URI` . The version segment is placed before module, controller, and method path fragments. [packages/core/router/route-path-factory.ts:27-57](#)
- A method-level version takes precedence over a controller-level version. [packages/core/router/route-path-factory.ts:80-84](#) Multiple versions generate one path per version. [packages/core/router/route-path-factory.ts:34-44](#)

- `VERSION_NEUTRAL` skips the version segment; in an array, it creates an additional unversioned path alongside versioned paths. [packages/core/router/route-path-factory.ts:38-51](#)
- The interface is one branch of `VersioningOptions`, which also includes the shared optional `defaultVersion`. [packages/common/interfaces/version-options.interface.ts:93-110](#) When versioning is configured, controller version metadata falls back to that shared default. [packages/core/router/routes-resolver.ts:225-235](#)
- `NestApplication.enableVersioning()` defaults to `{ type: VersioningType.URI }` when called without options and stores the supplied options in application configuration. [packages/core/nest-application.ts:288-293](#) [packages/core/application-config.ts:138-145](#)

No validation, normalization, or explicit error handling for `prefix` appears in the URI-prefix selection code; a defined string is returned as-is. [packages/core/router/route-path-factory.ts:86-96](#)